Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 894246, 5 pages
http://dx.doi.org/10.1155/2014/894246

Hindawi

Research Article

A Mahalanobis Hyperellipsoidal Learning Machine Class Incremental Learning Algorithm

Yuping Qin,[1] Hamid Reza Karimi,[2] Dan Li,[3] Shuxian Lun,[4] and Aihua Zhang[1]

[1] *College of Engineering, Bohai University, Jinzhou 121013, China*
[2] *Department of Engineering, Faculty of Engineering and Science, The University of Agder, 4898 Grimstad, Norway*
[3] *College of Mathematics and Physics, Bohai University, Jinzhou, China*
[4] *New Energy College, Bohai University, Jinzhou, China*

Correspondence should be addressed to Yuping Qin; qlq888888@sina.com

Received 23 December 2013; Accepted 31 December 2013; Published 11 February 2014

Academic Editor: Ming Liu

A Mahalanobis hyperellipsoidal learning machine class incremental learning algorithm is proposed. To each class sample, the hyperellipsoidal that encloses as many as possible and pushes the outlier samples away is trained in the feature space. In the process of incremental learning, only one subclassifier is trained with the new class samples. The old models of the classifier are not influenced and can be reused. In the process of classification, considering the information of sample's distribution in the feature space, the Mahalanobis distances from the sample mapping to the center of each hyperellipsoidal are used to decide the classified sample class. The experimental results show that the proposed method has higher classification precision and classification speed.

1. Introduction

Incremental learning is an intelligent technology of data mining and knowledge discovery. There are already some key methods of incremental learning, such as KNN, principal component analysis, Bayesian network, and Boosting and support vector machines (SVM), and this idea in the control theory, such as data driven [1–5], promotes the development of control method. Among these methods, SVM has a good generalization performance, because it does not depend on all the training data, but a subset named support vector. The number of support vectors is very small compared with the training data set, so SVM is a powerful tool to the incremental learning.

Many incremental learning algorithms based on SVM have been proposed, and they received better results, such as the incremental learning Batch SVM [6, 7], on-line recursive algorithm [8], divisional training SVM algorithm [9], fast incremental learning algorithm [10], and α-SVM algorithm [11]. However, these algorithms are only suitable to the case that the new incremental samples belong to old classes. When a new class was added to the classification system, the above methods could not be fully accommodated to this situation and the old models became useless. Reference [12] proposed a class incremental learning algorithm. The algorithm reuses the old models of the classifier and trains only one binary subclassifier when a new class comes. But it is not suitable to large data set, because all samples participate in the training in the process of each incremental learning. For the disadvantage, [13] proposed a class incremental learning algorithm based on hyper sphere support vector machine (HS-CIL), but the algorithm is only suitable to the case that the sample's distribution is hyper sphere shaped and the density is higher. For the disadvantage, [14] proposed a hyperellipsoidal class incremental learning algorithm (HE-CIL), but the algorithm does not consider the influence of the outlier samples. Therefore, a Mahalanobis hyperellipsoidal learning machine class incremental learning algorithm (MHE-CIL) is proposed in this paper. To every class sample, the smallest hyperellipsoidal that encloses as much samples as possible and pushes the outlier samples away is trained in the feature space. Mahalanobis distances are used to confirm the classified sample class.

The rest of this paper is organized as follows. In Section 2, a brief review of Mahalanobis hyperellipsoidal learning

machine is given. In Section 3, a new Mahalanobis hyperellipsoidal learning machine class incremental learning algorithm is discussed in detail. In Section 4, experimental results are given on Reuters 21578. Finally, conclusion is outlined in Section 5.

2. Mahalanobis Hyperellipsoidal Learning Machine

Given a set of training samples of a class $\{x_i\}_{i=1}^N$, where $x_i \in R^d$ and N is the number of samples, let X be a $d \times N$ sample matrix. Training a Mahalanobis hyperellipsoidal $E(a, R)$ in the feature space, where a is the center of the hyperellipsoidal and R is the radius of the hyperellipsoidal, the hyperellipsoidal encloses most of the mappings of sample and the radius R is as small as possible. If there are no remote points, then the hyperellipsoidal will enclose all the mappings of sample. If there are remote points, allowing part of the samples outside of the superellipsoid, and searching for the smallest superellipsoid which can surround the most samples. When we are uncertain whether there are remote points, nonnegative slack variables ξ_i $(i = 1, 2, \ldots, N)$ are introduced to allow some of the mappings of sample outside the hyperellipsoidal. Using the method that is similar to finding optimal hyperplane of SVM to obtain the smallest hyperellipsoidal [15–17], the formulation is as follows:

$$\begin{aligned} \min_{a,R,\xi_i} \quad & R^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & (x_i - a)^T\Sigma^{-1}(x_i - a) \le R^2 + \xi_i \\ & \xi_i \ge 0,\ i = 1, 2, \ldots, N, \end{aligned} \tag{1}$$

where C is used to compromise the number of noises out of hyperellipsoidal and the radius of hyperellipsoidal, Σ is covariance matrix of the samples and Σ^{-1} is the inverse of the covariance matrix Σ.

To solve the optimization problem above, one can construct the Lagrange function as follows:

$$\begin{aligned} L(R, a, \beta, \gamma, \xi_i) = {} & R^2 + C\sum_{i=1}^{N}\xi_i \\ & - \sum_{i=1}^{N}\alpha_i\left\{R^2 + \xi_i - (x_i - a)^T\Sigma^{-1}(x_i - a)\right\} \\ & - \sum_{i=1}^{N}\beta_i\xi_i, \end{aligned} \tag{2}$$

where $\alpha_i \ge 0$ and $\beta_i \ge 0$ are the Lagrange multipliers.

According to the Kuhn-Tucker theorem (KKT) in optimization theory, the following conditions are satisfied:

$$\begin{aligned} \frac{\partial L}{\partial R} &= 2R\left(1 - \sum_{i=1}^{N}\alpha_i\right) = 0 \Longrightarrow \sum_{i=1}^{N}\alpha_i = 1, \\ \frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \beta_i = 0 \Longrightarrow 0 \le \alpha_i \le C, \\ \frac{\partial L}{\partial a} &= -\sum_{i=1}^{N}2\alpha_i(x_i - a) = 0 \Longrightarrow a = \sum_{i=1}^{N}\alpha_i x_i. \end{aligned} \tag{3}$$

Substituting (3) into (2), the dual optimal problem is obtained as follows:

$$\begin{aligned} \max_{\alpha_i \ge 0} \quad & \sum_{i=1}^{N}\alpha_i x_i^T\Sigma^{-1}x_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j x_i^T\Sigma^{-1}x_j \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i = 1 \quad 0 \le \alpha_i \le C,\ i = 1, 2, \ldots, N. \end{aligned} \tag{4}$$

The kernel form of (4) is as follows:

$$\begin{aligned} \max_{\alpha_i \ge 0} \quad & \sum_{i=1}^{N}\alpha_i k(x_i, X)\, Q^T\Omega^{-2}Qk(x_i, X)^T \\ & - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j k(x_i, X)\, Q^T\Omega^{-2}Qk\left(x_j, X\right)^T \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i = 1 \quad 0 \le \alpha_i \le C,\ i = 1, 2, \ldots, N, \end{aligned} \tag{5}$$

where $K(\cdot)$ is kernel function and $K(x_i, x_j) = g(x_i)^T g(x_j)$.

The examples that lie outside or on the margin are the corresponding α_i nonzero. These examples are called support vectors.

The center of the smallest hyperellipsoidal a can be obtained as follows:

$$a = \sum_i \alpha_i g(x_i). \tag{6}$$

The square kernel Mahalanobis distance from the mapping $g(x)$ of sample x to the center a of the hyperellipsoidal in the feature space is defined as follows:

$$\begin{aligned} d^2(g(x), a) &= (g(x) - a)^T\Sigma^{-1}(g(x) - a) \\ &= N\left(k(x, X) - \sum_{i=1}^{N}\alpha_i k(x_i, X)\right)Q^T\Omega^{-2} \\ &\quad \times Q\left(k(x, X) - \sum_{i=1}^{N}\alpha_i k(x_i, X)\right). \end{aligned} \tag{7}$$

The radius of the smallest hyperellipsoidal R can be determined by (7), via KKT conditions as follows:

$$\begin{aligned} d^2(x,a) < R^2 \quad & \alpha_i = 0, \\ d^2(x,a) = R^2 \quad & 0 < \alpha_i < C, \\ d^2(x,a) > R^2 \quad & \alpha_i = C. \end{aligned} \tag{8}$$

3. Incremental Learning Algorithm

We give a set of training samples $A = \{x_i, E_i\}_{i=1}^{l}$ and kernel function $K(\cdot)$, where $x_i \in R^n$, $E_i \in \{1, 2, \ldots, N\}$, N is the number of class of the training set A, l is the number of training samples, and K corresponds to inner product in feature space, namely, $K(x_i, x_j) = g(x_i)^T g(x_j)$.

Assume that A^m is a subset of training samples A, where all the samples of the subset A^m belong to the mth class ($m = 1, 2, \ldots, N$). For every subset A^m, the smallest hyperellipsoidal $E(a_m, R_m)$ is trained in feature space, where a_m is the center of the hyperellipsoidal and R_m is the radius of the hyperellipsoidal.

If a new class, which defined B, is generated, training the smallest hyper ellipsoidal $E(a_{N+1}, R_{N+1})$ in the feature space, and adding the classifier to old models. One time incremental learning is finished.

For the sample x to be classified, compute the Mahalanobis distance from the mapping $g(x)$ of the sample x to the center a_m of the hyperellipsoidal according to (7).

If all $d_m(g(x), a_m)$ satisfy $d_m(g(x), a_m) > R_m$ ($m = 1, 2, \ldots, N$), and there is no hyperellipsoidal that encloses the mapping $g(x)$ of the sample x, then compute the membership that the sample x belongs to the mth class according to (9) and then confirm the class of sample x as follows:

$$r_m = \frac{R_m}{d_m(g(x), a_m)}, \tag{9}$$

$$\text{class} = \arg\max_i r_i. \tag{10}$$

If there are no less than two $d_m(g(x), a_m)$ ($m \in \{1, 2, \ldots, N\}$) that satisfy $d_m(g(x), a_m) \le R_m$, then compute the membership that the sample x belongs to the mth class according to (11) and then confirm the class of sample x as follows:

$$r_i = 1 - \frac{d(g(x), a_m)}{R_m}. \tag{11}$$

For the sample x to be classified, the classification algorithm is described in detail as follows.

Step 1. Computing $d_m(g(x), a_m)$ ($m = 1, 2, \ldots, N$) according to (7).

Step 2. If only one $d_m(g(x), a_m)$ ($m \in \{1, 2, \ldots, N\}$) satisfies $d_m(g(x), a_m) \le R_m$, then the sample belongs to the mth class; go to Step 5; otherwise go to Step 3.

Table 1: Training set and testing set.

Sample	Wheat	Corn	Coffee	Soybean	Cocoa
Class code	1	2	3	4	5
Training set	204	168	97	79	50
Texting set	101	84	48	40	25

Step 3. If all $d_m(g(x), a_m)$ ($m = 1, 2, \ldots, N$) satisfy $d_m(g(x), a_m) > R_m$, then compute the membership that the sample x belongs to the mth class according to (9) and then confirm the class of sample x by (10) and go to Step 5; otherwise go to Step 4.

Step 4. If more than one $d_m(g(x), a_m)$ ($m \in \{1, 2, \ldots, N\}$) satisfies $d_m(g(x), a_m) \le R_m$, then compute the membership that the sample x belongs to the mth class according to (11) and then confirm the class of sample x by (10).

Step 5. End.

4. Experiments

Experiments are made on Reuters 21578, in which five categories and 896 texts are used. 598 texts are used as training set, and the rest 298 texts are used as testing set (see Table 1). Information gain is used to reduce feature dimension and the weight of every word is computed according to TF-IDF.

To verify the efficiency of the proposed method, the same tasks are realized by using HS-CIL, HE-CIL, and MHE-CIL methods. The computational experiments were done on a Pentium 1.6 G with 512 MB memory. Liner kernel function is used for all the experiments. System parameter $C = 10$, $\nu = 0.01$.

The macroaverage precision (MAAP), macro average recall (MAAR), and macroaverage F_1 (MAAF) [18] are used to evaluate the classification performance of the algorithm.

In experiments, the original sample set includes two class samples (wheat and corn). Three times class incremental learning is done; the first time increment is the third class sample (coffee), the second time increment is the forth class sample (soybean), and the third time increment is the fifth class sample (cocoa). The macroaverage precision, macroaverage recall, and macroaverage F_1 value of three algorithms are given in Table 2. The training time and testing time of three algorithms are given in Table 3.

The experimental results show the precision, the recall, and the F_1 value of MHE-CIL method which are obviously higher than the other two methods. The key reasons are that MHE-CIL method reduces the space that hyperellipsoidal encloses by pushing the outlier samples away, and the information of sample's distribution is considered by using Mahalanobis distance. The training time of MHE-CIL method is faster compared with HE-CIL method, and it is nearly the same as HS-CIL method. The classification speed of MHE-CIL method is faster compared with HE-CIL method, and it is nearly the same as HE-CIL method.

Table 2: The comparison of MAAP, MAAR, and MAAF.

Learning	Algorithm	MAAP (%)	MAAR (%)	MAAF (%)
Original	HS-CIL	72.76	72.84	72.79
	HE-CIL	76.92	74.52	75.18
	MHE-CIL	78.76	76.64	77.82
1st increment	HS-CIL	79.75	78.09	78.85
	HE-CIL	80.00	78.53	79.26
	MHE-CIL	81.75	79.73	80.39
2nd increment	HS-CIL	74.97	71.53	73.21
	HE-CIL	82.56	78.72	80.11
	MHE-CIL	83.72	79.63	81.66
3rd increment	HS-CIL	76.64	73.47	75.02
	HE-CIL	81.41	76.61	78.40
	MHE-CIL	83.56	78.31	81.84

Table 3: The comparison of training time and testing time.

Learning	Algorithm	Training times (ms)	Testing times (ms)
Original	HS-CIL	110	94
	HE-CIL	126	83
	MHE-CIL	112	88
1st increment	HS-CIL	15	140
	HE-CIL	33	132
	MHE-CIL	16	135
2nd increment	HS-CIL	32	157
	HE-CIL	36	161
	MHE-CIL	33	158
3rd increment	HS-CIL	31	204
	HE-CIL	48	185
	MHE-CIL	32	187

5. Conclusion

A novel class incremental learning algorithm is proposed. In the process of class incremental learning, only the new class samples participate in training and the old models of the classifier can be reused. In the process of classification, the Mahalanobis distance is used to confirm the class of classified sample, and the information of the sample's distribution in the feature space is considered. The experimental results show that the proposed algorithm not only improves classification accuracy obviously, but also ensures training speed and classification speed. How to use kernel function theory to increase the density of the samples would be our research work in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study is partly supported by the National Natural Science Foundation of China (no. 61304149), the Natural Science Foundation of Liaoning Province in China (no. 201202003), and the Program for New Century Excellent Talents in University (no. NCET-11-1005).

References

[1] S. Yin, S. X. Ding, A. H. A. Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.

[2] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[3] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2014.

[4] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, 2013.

[5] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.

[6] S. Rüping, "Incremental learning with support vector machines," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '01)*, pp. 641–642, December 2001.

[7] C. Domeniconi and D. Gunopulos, "Incremental support vector machine construction," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '01)*, pp. 589–592, December 2001.

[8] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, pp. 409–415, 2001.

[9] J. Zhang, Z. Li, and J. Yang, "A divisional incremental training algorithm of Support Vector Machine," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '05)*, pp. 853–856, August 2005.

[10] R. Kong and B. Zhang, "Fast incremental learning algorithm for support vector machine," *Control and Decision*, vol. 20, no. 10, pp. 1129–1136, 2005.

[11] R. Xiao, J. Wang, and F. Zhang, "An approach to incremental SVM learning algorithm," in *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '00)*, pp. 268–273, 2000.

[12] B.-F. Zhang, J.-S. Su, and X. Xu, "A class-incremental learning method for multi-class support vector machines in text classification," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2581–2585, August 2006.

[13] Y. P. Qin, X. N. Li, and C. l. Wang, "Study on class incremental learning algorithm based on hyper-sphere support vector machines," *Computer Science*, vol. 8, article 28, 2008.

[14] A. Ghorbel, A. Almaksour, A. Lemaitre, and E. Anquetil, "Incremental learning for interactive sketch recognition," in *Graphics Recognition: New Trends and Challenges*, pp. 108–118, Springer, New York, NY, USA, 2013.

[15] G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali, "Hacking smart machines with smarter ones: how to extract meaningful data from machine learning classifiers," *CoRR*, http://arxiv.org/abs/1306.4447.

[16] F. Porikli and Y. Chi, "Multi-class classification method," US Patent 20, 130, 156, 300, 2013.

[17] N. Shahid, I. H. Naqvi, and S. B. Qaisar, "One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments," *Artificial Intelligence Review*, 2013.

[18] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.